1. Report No. NASA TM X-3572	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle FLUID: A NUMERICAL INTERPOLATION PROCEDURE FOR OBTAINING THERMODYNAMIC AND TRANSPORT PROPERTIES OF FLUIDS		5. Report Date August 1977
		6. Performing Organization Code
7. Author(s) Theodore E. Fessler		8. Performing Organization Report No. E-9132
		10. Work Unit No. 505-01
9. Performing Organization Name and Address National Aeronautics and Space Administration Lewis Research Center Cleveland, Ohio 44135		11. Contract or Grant No.
		13. Type of Report and Period Covered Technical Memorandum
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546		14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

A computer-program subroutine, FLUID, has been developed to calculate thermodynamic and transport properties of pure fluid substances. It provides for determining the thermodynamic state from assigned values for temperature-density, pressure-density, temperature-pressure, pressure-entropy, or pressure-enthalpy. FLUID considers liquid or two-phase (liquid-gas) conditions as well as the gas phase. A van der Waals model is used to obtain approximate state values; these values are then corrected for real-gas effects by model-correction factors obtained from tables based on experimental data. Saturation condition, specific heat, entropy, and enthalpy data are included in the tables for each gas. Since these tables are external to the FLUID subroutine itself, FLUID can implement any gas for which a set of tables has been generated. (A setup phase is used to establish pointers dynamically to the tables for a specific gas.) Data-table preparation is described. FLUID is available in both SFTRAN and FORTRAN.

17. Key Words (Suggested by Author(s)) Thermodynamic properties; Thermophysical properties; Real gases; Computer programs; Vapor phases; Liquids	18. Distribution Statement Unclassified - unlimited STAR Category 61		
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 25	22. Price* A02

* For sale by the National Technical Information Service, Springfield, Virginia 22161

FLUID: A NUMERICAL INTERPOLATION PROCEDURE FOR OBTAINING THERMODYNAMIC AND TRANSPORT PROPERTIES OF FLUIDS

by Theodore E. Fessler

Lewis Research Center

SUMMARY

A computer-program subroutine, FLUID, has been developed to calculate the thermodynamic and transport properties of pure fluid substances. It provides for determining the thermodynamic state from assigned values for temperature-density, pressure-density, temperature-pressure, pressure-entropy, or pressure-enthalpy. FLUID considers liquid or two-phase (liquid-gas) conditions as well as the gas phase. A van der Waals model is used to obtain approximate state values; these values are then corrected for real-gas effects by model-correction factors obtained from tables based on experimental data. Saturation conditions, specific heat, entropy, and enthalpy data are included in the tables for each gas. Since these tables are external to the FLUID subroutine itself, FLUID can implement any gas for which a set of tables has been generated. (A setup phase is used to establish pointers dynamically to the tables for a specific gas.) Data-table preparation is described. FLUID is available in both SFTRAN and FORTRAN.

INTRODUCTION

For several years the computer code GASP (ref. 1) has been used at the Lewis Research Center to supply thermodynamic and transport property values for engineering calculations. Because current and projected use of GASP is so great, the replacement code FLUID has been developed as a cost-saving measure. FLUID provides thermodynamic and transport properties in good agreement with GASP and WASP and some savings in computer time.

GASP is a FORTRAN subprogram, the current version of which has been implemented for 10 gases: hydrogen, oxygen, nitrogen, argon, carbon dioxide, carbon monoxide, fluorine, helium, neon, and methane. A similar program, WASP (ref. 2), is used for water or steam properties. In GASP, a modified virial equation of state is used

to express pressure as a function of temperature and density. In WASP, thermodynamic properties are calculated from the Helmholtz free-energy function. In both programs, transport properties are calculated from empirical functions in temperature and density. GASP and WASP are applicable to both the liquid and gas phases, including saturation conditions in which a mixture of phases occurs.

All of the thermodynamic properties can be obtained from an equation of state (or from the Helmholtz function in the case of WASP). Hence, in principle, all of the thermodynamic property data for one gas can be combined by a constrained, weighted least-squares technique in which the equation-of-state coefficients are obtained numerically. The coefficients used in GASP (and WASP) were obtained in this way.

A disadvantage to this approach is that it is hard to find a functional form for the equation of state (or the Helmholtz function) that works well over the entire temperature-density plane. Many terms are required. And, it is costly to solve these functions inversely - for independent variables in terms of dependent variables. For example, GASP takes much longer to compute temperature and density from pressure and enthalpy than it does to compute pressure and enthalpy from temperature and density. Another problem may occur if the choice of the algorithm used to obtain these inverse solutions depends on the argument values. If that be the case, not only will the program be more complex, but there may be discontinuities in calculated values for the different algorithms.

In the FLUID subroutine, using a simple gas model and empirical corrections to it, obtained by numerical interpolation, avoids some of these complexities and makes possible some savings in computer time.

SYMBOLS

A	sonic velocity
a	tabulated argument values
C	specific heat
$C_1, \ldots, C_8$	constants used in data tables
c	coefficients used in interpolation formulas
d	reduced density, ρ/ρ_c
F, G	tabular functions
H	enthalpy
P	pressure
p	reduced pressure, P/P_c

2

R	gas constant
S	entropy
T	temperature
t	reduced temperature, T/T_c
U	internal energy
V	specific volume
W	mass fraction of one phase in a two-phase system
x	argument of interpolation
$Z_1, \ldots, Z_6$	model correction factors
α, β	constants
γ	ratio of specific heats, C_p/C_v
ρ	density

Subscripts:

c	value at critical point
g	gas-phase value
l	liquid-phase value
p	at constant pressure
t	at constant temperature
v	at constant volume
0	function of temperature for ideal gas

PROGRAM DESIGN

FLUID, like GASP, is designed to calculate thermodynamic and transport properties of pure fluids in both the liquid and gas phases. In FLUID, an equation of state provides a model for calculating thermodynamic properties; tables of model-correction factors are used to bring the equation of state into agreement with accepted values, as for example, those from GASP. Other tables similar to the model-correction-factor tables are used directly, without a model, to calculate viscosity and thermal conductivity. Interpolation formulas are used for argument values between those in the tables.

Two auxiliary subroutines are used by the FLUID subroutine: NTRP, in which all interpolations are performed, and CUBIC, in which the equation of state is solved for density.

The tables for each gas are external to FLUID. New gases may be implemented by building the appropriate tables from an already existing code like GASP or by some direct use of tabulated values. The tables of thermodynamic properties are in nondimensional form to allow for a change in the system of units.

<center>Equation of State</center>

The van der Waals equation of state can qualitatively represent the behavior of a real fluid in both its liquid and gaseous states. Hence, it has obvious application in the work at hand.

Model. - In its reduced form, van der Waals' equation is

$$\frac{(p + 3d^2)(3 - d)}{8td} = 1$$

(See SYMBOLS.) Real liquids and gases show a behavior somewhat different from that predicted by van der Waals' equation. We choose to account for these differences by using a correction factor Z_1, where

$$\frac{(p + 3d^2)(3 - d)}{8td} = Z_1(t, d)$$

which is found experimentally to be a smooth, slowly varying function of temperature and density. (Similar methods have been used by others to obtain correction factors, residual values, or excess functions that relate observed property values to values predicted by a model. See, e.g., ref. 3.) If values of Z_1 are available on a coarse tabular grid in t and d, two-dimensional interpolation can be used to obtain Z_1 for intermediate values of t and d.

The behavior of real fluids will also depart from van der Waals' model whenever two-phase conditions occur. In that case, pressure, liquid-phase density, and gas-phase density are expressible as functions of temperature only. These functions can be arranged in tabular form so that interpolation can supply function values at intermediate argument values.

Limitations of model. - Although the van der Waals equation gives a satisfactory qualitative representation of the behavior of a real gas, it fails in one important aspect for the present application. This failure comes from the derivation of the model, in

4

which it is assumed that molecules have some hard, nonzero volume. That is, there is a certain density that cannot be exceeded: three times the critical density. But some real fluids have liquid densities that are greater than this model maximum. For example, the density of water at room temperature is 3.15 times the critical density. This limits the liquid-phase range of FLUID to temperatures greater than 400 K for water (although gas and two-phase calculations can be performed down to 273 K).

One way around this limitation is to assume a somewhat higher, pseudocritical density, both when the model correction tables are generated and when they are used. For water, a pseudocritical density of 0.34 g/cm^3 instead of the observed value of 0.317 could be used. This has been found to work well, except that the resulting tables are not to be used near the critical point. In this example for water, this would be the region in which 350 K $<$ T $<$ 410 K and 0.25 g/cm^3 $< \rho <$ 0.40 g/cm^3. Fortunately, many fluids of interest are not so seriously affected by this density limitation as is water.

Other Thermodynamic Functions

The van der Waals equation of state implies certain behavior of other thermodynamic variables and fluid properties as well. Since the specific heat at constant volume satisfies

$$\frac{1}{T}\left(\frac{\partial C_v}{\partial V}\right)_t = \left(\frac{\partial^2 P}{\partial T^2}\right)_v$$

it follows that, for a van der Waals gas, C_v must be at most a function of temperature:

$$C_v = C_{v,0}(t)$$

Also, since

$$C_p - C_v = T\left(\frac{\partial P}{\partial T}\right)_v \cdot \left(\frac{\partial V}{\partial T}\right)_p$$

it can be shown that

$$C_p - C_v = \frac{R}{1 - \frac{d(3-d)^2}{4t}}$$

Then, since entropy can be expressed as

$$S = \int \frac{C_v}{T} dT + \int \left(\frac{\partial P}{\partial T}\right)_V dV + \text{Constant}$$

it follows that

$$S = S_0(t) + R \ln \left(\frac{3-d}{d}\right)$$

where $S_0(t)$ is a function of temperature only for an ideal gas. Likewise, internal energy can be expressed as

$$U = \int C_v \, dT + \int \left[T \left(\frac{\partial P}{\partial T}\right)_V - P \right] dV + \text{Constant}$$

which leads to

$$U = U_0(t) - \frac{9}{8} RT_c d$$

And, from this result and $H = U + PV$, we can obtain

$$H = U_0(t) + \frac{9}{8} RT_c \left(\frac{p}{3d} - d\right)$$

Finally, the sonic velocity in a van der Waals' gas can be derived from

$$A^2 = \gamma \left(\frac{\partial P}{\partial \rho}\right)_T$$

to be

$$A = \sqrt{\frac{3}{8} \gamma RT_c \left[\frac{8t}{3-d} \left(1 + \frac{d}{3-d}\right) - 6d \right]}$$

where $\gamma = C_p/C_v$.

These equations only approximate the behavior of real liquids and gases. As in the case of the equation of state, we choose to account for these differences by the use of

6

correction factors, Z_i, which can be put in tabular form. The following relationships define these correction factors:

$$S = Z_2(t, d) \times \left[S_0(t) + R \ln \left(\frac{3 - d}{d} \right) \right]$$

$$H = Z_3(t, d) \times \left[U_0(t) + \frac{9}{8} RT_c \left(\frac{p}{3d} - d \right) \right]$$

$$C_v = Z_4(t, d) \times C_{v,0}(t)$$

$$C_p = Z_5(t, d) \times \left[C_v + \frac{R}{1 - \dfrac{d(3 - d)^2}{4t}} \right]$$

$$A = Z_6(t, d) \sqrt{\frac{3}{8} \gamma RT_c \left[\frac{8t}{3 - d} \left(1 + \frac{d}{3 - d} \right) - 6d \right]}$$

If saturation occurs, two-phase conditions prevail and we can express liquid- and gas-phase values of entropy and enthalpy as functions of temperature only. (We do not define specific heats or sonic velocity when two phases occur.) The entropy and enthalpy of the mixture are then given by

$$S = W_l S_l + W_g S_g$$

$$H = W_l H_l + W_g H_g$$

where the subscripts l and g refer to the liquid and gas phases, respectively, and W is a mass fraction in one phase.

Method of Interpolation

Interpolation is used in this package to obtain values of functions and correction factors at points intermediate to tabulated values. The method used has been called the "osculatory" or "double-Lagrangian" interpolation method. Following Lagrange's three-point interpolation formula, we find the value of a function by

$$F(x) = \frac{(x - a_2)(x - a_3)}{(a_1 - a_2)(a_1 - a_3)} F(a_1) + \frac{(x - a_1)(x - a_3)}{(a_2 - a_1)(a_2 - a_3)} F(a_2) + \frac{(x - a_1)(x - a_2)}{(a_3 - a_1)(a_3 - a_2)} F(a_3)$$

where x is an argument intermediate between the tabular values a_1, a_2, and a_3 and $F(a_1)$, etc., are the tabulated values of the function on the a grid. Suppose x lies between a_2 and a_3. Then $F(x)$ could also be calculated by using tabular values at a_2, a_3, and a_4 (if a_4 exists). Let the formula using a_1, a_2, and a_3 be called LOWER(x) and the formula using a_2, a_3, and a_4 be called UPPER(x). Then by double interpolation,

$$G(x) = \frac{(a_3 - x)}{(a_2 - a_3)} \text{LOWER}(x) + \frac{(x - a_2)}{(a_3 - a_2)} \text{UPPER}(x)$$

in which a weighted average of the two interpolations is prescribed. This method of interpolation results in a very smooth function because $G(x)$ has a continuous first derivative.

The interpolation formulas just described can be generalized as

$$G(x) = \sum_{i=1}^{n} c_i \times F(a_i)$$

where $n = 3$ for the three-point formula and $n = 4$ for the weighted-average formula. The values of c_i depend on where x falls in the table of arguments but not on the function values in the table. Thus, if several tabular functions share the same argument table, the same interpolation coefficients apply equally to all functions for any given value of x. In order to take advantage of this, calls to the interpolation subprogram NTRP are made in two steps: first, the coefficients c_i are calculated and saved and then, for each tabular function to be interpolated, a call is made with that function table as the argument. If the argument value x falls in the argument table so that $a_2 \leq x < a_{m-1}$, where m is the length of the argument table, the weighted-average formula can be used. If not, the three-point formula is used.

For interpolation in two dimensions, the interpolation formula can be expressed as a double summation over n^2 tabulated values. In this case, each argument table has associated with it a set of c_i values that are calculated by using these formulas.

8

Solutions for Other Independent Variables

Three sets of tabular data are used in this package: (1) the ideal-gas functions $S_0(t)$, $U_0(t)$, and $C_{v,0}(t)$; (2) saturation values $p(t)$, $d_l(t)$, $d_g(t)$, $S_l(t)$, $S_g(t)$, $H_l(t)$, and $H_g(t)$; and (3) correction factors $Z_1(t,d)$. Interpolation in these tables provides all data necessary to calculate thermodynamic properties for a given temperature and density. But more generally, since any two thermodynamic variables determine the fluid state, we require solutions for those cases where the given variables may be other than t and d. Four of these cases have been implemented and will now be considered individually.

Pressure and density. - When density is an input parameter, a preliminary calculation must be made to determine if two-phase conditions prevail. If the pressure is less than critical, the liquid- and gas-phase densities can be obtained from the saturation tables by interpolation. A two-phase condition is indicated when $d_g < d < d_l$. If that is the case, the liquid- and gas-phase values of the other thermodynamic variables can also be obtained by interpolation of the saturation tables.

If only one phase exists, the temperature is found by successive approximation, with $Z_1 = 1.0$ as a starting value. Iteration continues until temperature differences are smaller than a convergence criterion, C_5.

Temperature and pressure. - If the temperature is less than critical, van der Waals' equation yields three roots when solved for density. The smallest of these corresponds to the gas value, and the largest corresponds to the liquid value. (These values are calculated in subroutine CUBIC.) If the pressure is less than saturation, the gas value is used; otherwise, the liquid value is used. Van der Waals' equation yields only one root (gas) when the temperature is greater than critical.

In either case, density is found by successive approximation, with $Z_1 = 1.0$ as a starting value. Iteration continues until density differences are smaller than a convergence criterion, C_6.

Pressure and entropy. - When entropy is an input parameter, a preliminary calculation is made to decide if two-phase conditions prevail. If the pressure is less than critical, the liquid and gas entropies can be obtained from the saturation tables by interpolation. A two-phase condition is indicated when $S_l < S < S_g$. If that is the case, the liquid- and gas-phase values of the other thermodynamic variables can also be obtained by interpolation of the saturation tables.

If only one phase exists, density and temperature are found by successive approximation. If the given pressure is less than critical, the liquid or gas saturation density is used as a starting value, depending on whether the entropy is less than or more than critical. If the pressure is more than critical, the calculation starts at critical density. By using the procedure described for pressure and density, trial values of density lead to trial values of temperature.

The entropy calculated from trial values of temperature and density is then compared with the given entropy, and a new trial value of density is computed by using an approximation for entropy as a function of density. If the fluid state is gas-like (entropy greater than critical), this approximation is of the form $S(d) = \alpha + \beta \log(d)$, where α and β are constants; if the fluid state is liquid-like, the approximation is of the form $S(d) = \alpha + \beta \log (3 - d)$. Iteration continues until entropy differences satisfy a convergence criterion, C_7.

Pressure and enthalpy. - The method used here is similar to that for the case of pressure and entropy; a preliminary calculation detects two-phase conditions and, if that is the case, the other thermodynamic variables are obtained by interpolation of the saturation tables.

If only one phase exists, successive approximation is used to solve for temperature and density. The fluid state is considered gas-like if the given enthalpy is more than the saturated-liquid enthalpy (and the pressure is less than critical) or if the given enthalpy is more than the critical-point enthalpy (and the pressure is greater than critical). The approximations for enthalpy as a function of density are of the form $H(d) = \alpha + \beta/d$ if gas-like and $H(d) = \alpha + \beta d$ if liquid-like. Iteration continues until enthalpy differences satisfy a convergence criterion, C_8.

Calling Conventions for FLUID

FLUID, NTRP, and CUBIC were coded in SFTRAN (Structured FORTRAN), a programming language recently introduced at the Lewis Research Center (ref. 4). This language was chosen for its self-documenting features. It allows and even encourages the grouping of instructions into small, natural units that can be given unique, descriptive names. In listings, these are displayed in a manner that makes their internal structure immediately apparent to the eye.

Other than the manner in which branching and looping are handled, SFTRAN is very much like FORTRAN; SFTRAN routines can call, or be called by, FORTRAN routines. The SFTRAN language has been implemented as a precompiler that generates FORTRAN source code from SFTRAN source code. The FORTRAN versions of FLUID, NTRP, and CUBIC so generated have been included in the FLUID package for use by those not having an SFTRAN precompiler. (This FORTRAN code will not be helpful in understanding FLUID; the FORTRAN produced by the SFTRAN precompiler is hard to read.)

Initializing FLUID for a new gas. - FLUID does not contain any data tables. Instead, all of the tables for one gas are collected into a gas-tables subroutine (one subroutine for each gas). Then, when the user's program is ready to calculate properties for a particular gas, it calls the gas-tables subprogram for that gas, which in turn calls FLUID0,

setting up pointers to the appropriate tables. (See appendix A for a description of the entries in FLUID.) Subsequent calls to FLUID for property values will then use these tables until such time as a different gas-tables subprogram is called by the user's program.

A library of gas-tables subroutines has been developed. Each subroutine in the library has been named for the particular gas it implements. Hence, O2 is for oxygen, PH2 is for para-hydrogen, etc. These subroutines are available in source code and can be transported to other computers.

<u>Calls to FLUID for property values.</u> - Five combinations of input variables have been implemented: (T, ρ), (P, ρ), (T, P), (P, S), and (P, H). A particular one is specified by integer ENTRY. Temperature, pressure, and density are transmitted explicitly as calling arguments. Entropy and enthalpy are transmitted in the array PROPS, along with the specific heats, sonic velocity, and the transport properties. An integer, NP, is used to specify the length of PROPS and hence how many of these properties are to be calculated.

As an example, consider that at some point in the user's program, a calculation of oxygen properties is required. In particular, suppose it is desired to obtain the temperature, density, and specific heats at constant volume from pressure and enthalpy. For this, the calling program might contain the sequence

        CALL O2
        NPROPS=4
        ENTRY=5
        PROPS(3)=H
        CALL FLUID (T, P, D, PROPS, NPROPS, ENTRY, VAPOR, ERROR)
        IF (ERROR. NE. 0) DO (REPORT ERROR IN FLUID)
        IF (VAPOR) THEN
            DO (SPECIAL CASE: MIXED-PHASE CONDITION)
            ELSE
            S=PROPS(2)
            CV=PROPS(4)
            DO (NORMAL-CASE CALCULATION)
        END

(It is assumed in this example that ENTRY and ERROR have been specified as INTEGER type and VAPOR as LOGICAL type and that PROPS is a dimensioned array.) In this example, the highest index in PROPS of a desired property is 4 (specific heat at constant volume), and so the number of properties requested from FLUID need be only 4. The remaining properties (5 to 8) would not be calculated, thereby effecting a saving of computer time. The reader is referred to appendix B for more examples of the use of FLUID.

If the values of the input variables correspond to a mixed-phase condition, VAPOR

will be .TRUE. In that case, specific heats, sonic velocity, and transport properties are not calculated but liquid- and gas-phase values of mass fraction, density, entropy, and enthalpy may be available in the COMMON block/FLUIDC/depending on how many properties (NP) have been asked for. Liquid- or gas-phase values of specific heats, sonic velocity, and transport properties can be obtained by the auxiliary entries FLUIDL and FLUIDG.

Error reports. - FLUID recognizes certain kinds of errors in calling arguments. Fatal errors (due to an error in coding in the calling program) are announced on I/O UNIT 6 and result in a program STOP. Errors due to one or more of the table arguments being out of range are not fatal; they are signaled by certain bits in integer ERROR. If for some reason, convergence is not achieved in solving for temperature and/or density from other input variables, this condition is also signaled by a bit in ERROR.

GENERATION OF TABLES

If all, or nearly all, of the property data for a particular fluid are available in a program such as GASP, making tables for FLUID can be a programmed operation. On the other hand, it is possible (but with a greater effort) to make the desired tables directly from tabular data. In the example used here, tables for oxygen were produced from data partly from GASP and partly from NBS Circular 564 (ref. 5).

Gas Constant, Critical Values, and Convergence Constants

The following values were used for the oxygen tables:

$$R \ (C_1) = 0.25983 \ J/g\text{-}K \qquad C_5 = 0.001$$
$$T_c \ (C_2) = 154.78 \ K \qquad C_6 = 0.002$$
$$P_c \ (C_3) = 5.083 \ MN/m^2 \qquad C_7 = 0.001$$
$$\rho_c \ (C_4) = 0.4325 \ g/cm^3 \qquad C_8 = 0.001$$

The dimensions used for R, T, P, and ρ also determine the dimensions used for entropy, enthalpy, specific heats, and sonic velocity. For the dimensions indicated, entropy and specific heats will be in J/g-K, enthalpy will be in J/g, and sonic velocity will be in J/g to the 1/2 power. (If the more usual units of m/sec are desired for sonic velocity, the user should multiply the value returned from FLUID by 158.33.) The four convergence constants required by FLUID (C_5 to C_8) were determined by trial. Their values are not critical. The particular values chosen give good accuracy and at the same time minimize the number of iterations required to obtain that accuracy.

12

Saturation Tables

FLUID does not make use of a theoretical vapor-pressure equation such as the Clausius-Clapeyron equation. Instead, it uses the interpolation formulas described earlier. Since these formulas are cubic in the argument, they do not approximate well the vapor-pressure and vapor-density data. But if temperature arguments for the saturation tables are chosen to make each vapor-pressure value about twice that of the preceding value, the interpolated vapor pressures will track the correct function within about 1 part per thousand. (The interpolation algorithm requires temperature arguments in all tables to be in ascending order, but they need not have equal intervals. This requirement is imposed by the table-lookup procedure used in subroutine NTRP.) The reduced-temperature argument values used for oxygen are 0.42, 0.43, 0.45, 0.475, 0.5, 0.53, 0.56, 0.60, 0.65, 0.70, 0.75, 0.80, 0.85, 0.90, 0.95, 0.97, 0.982, 0.99, 0.995, 0.998, 0.999, and 1.0.

Inasmuch as entropy and enthalpy are defined as integrals, a reference value may be assigned arbitrarily to each. But for FLUID, it is necessary that entropy and enthalpy be positive in the temperature and density range covered by the model-correction tables. Otherwise, the use of model-correction factors to obtain real-gas values of entropy and enthalpy will not work. This requirement is satisfied by using reference values that make the saturated-vapor entropy and enthalpy twice their saturated-liquid values at the lowest temperature in the saturation tables.

The last value of each saturation-table quantity is the critical value. The critical values of reduced temperature, density, and pressure are unity, of course; the critical entropy and critical enthalpy are calculated as the arithmetic average of the next-to-last liquid- and gas-phase values.

Tables in Temperature Only

FLUID requires tables of values for the functions $S_0(t)$, $U_0(t)$, and $C_{v,0}(t)$, together with their argument values. According to the procedure discussed in the section Other Thermodynamic Functions, these ideal-gas functions are evaluated for some low-density value by

$$S_0(t) = S(t,d) - R \ln\left(\frac{3-d}{d}\right)$$

$$U_0(t) = H(t,d) - \frac{9}{8}RT_c\left(\frac{p}{3d} - d\right)$$

$$C_{v,0}(t) = C_v(t,d)$$

in which the S, H, and C_v values are obtained from the data base. (The density chosen should be low enough that saturation conditions are avoided over the entire temperature range used.)

The values of the argument t used for these tables should be selected to give good agreement between interpolated values and measured values. If specific heat shows marked variations with temperature, it may be necessary to use finer spacing where that occurs. The reduced-temperature arguments used for oxygen are 0.42, 0.5, 0.6, 0.72, 0.85, 0.98, 1.1, 1.25, 1.5, 1.75, 2.0, 2.5, 3.0, 3.7, 4.75, 6.0, 7.2, and 8.0.

The version of GASP available at the time of this writing implements oxygen data up to 500 K (reduced temperature of 3.23). It was desired to extend FLUID's capability for oxygen to a reduced temperature of 8.0 (1238 K). To this end, the $C_{v,0}(t)$ table was extended by using the 0.01-atmosphere specific heats for oxygen of reference 5 above 300 K. Then the $S_0(t)$ values were calculated by numerical integration of

$$\frac{S_0(t_2,d)}{R} = \frac{S_0(t_1,d)}{R} + \int_{t_1}^{t_2} \frac{C_{v,0}}{Rt} \, dt$$

and the $U_0(t)$ values were calculated by numerical integration of

$$\frac{U_0(t_2,d)}{RT_c} = \frac{U_0(t_1,d)}{RT_c} + \int_{t_1}^{t_2} \frac{C_{v,0}}{R} \, dt + \left(\frac{P_c}{\rho_c T_c R} - \frac{3}{8}\right)\left[\frac{p(t_2) - p(t_1)}{d}\right]$$

(For an ideal van der Waals' gas, the last term on the right is zero; for oxygen, it was calculated by using the van der Waals model and the low-density Z_1 value for oxygen of 1.2832.)

Tables in Temperature and Density

The t-d tables make up the bulk of the data used by FLUID (92 percent in the case of oxygen). Some care is needed in selecting the values for temperature and density if table size is to be minimized while maintaining good agreement with the data base over the entire temperature-density plane. The values used for oxygen are given in table I.

TABLE I. - REDUCED-TEMPERATURE

AND REDUCED-DENSITY ARGUMENT

VALUES USED IN TWO-ARGUMENT

TABLES FOR OXYGEN

Reduced temperature, t	Reduced density, d
0.42	0.0001
.45	.0003
.5	.0007
.59	.002
.68	.01
.8	.03
.92	.1
.985	.3
1.05	.6
1.15	1.0
1.3	1.45
1.6	1.85
2.0	2.2
2.6	2.46
3.5	2.65
4.6	2.79
6.0	2.87
8.0	2.92

These values were arrived at by a trial-and-error process in which some general guides were developed. For instance, it was found best to have adjacent argument-intervals in density differ by at most a factor of 3. Also, it was found helpful to have some temperature-density points distributed more or less uniformly alongside the saturation curve. Thus, at $t = 0.42$, the points at $d = 0.0003$ (gas) and $d = 2.92$ (liquid) lie just outside the two-phase region; at $t = 0.5$, the points at $d = 0.0007$ and $d = 2.87$ lie just outside the two-phase region, etc. For other gases, circumstances may well require a different mesh; table I should be used only as a starter.

Model-correction factors. - Even though GASP was not implemented for oxygen at temperatures above 500 K, it produced Z_1 values that were in agreement with reference 5 up to 1238 K. (This is not too surprising; none of the Z_1 values varies more

than about 0.002 from 500 K to 1238 K for pressures below 30 MN/m^2.) Hence, Z_i values for the entire table were based on values calculated by using GASP. At some high-density, high-temperature points, the GASP values were obviously erroneous. (These points were all beyond the pressure range of the data on which GASP is based.) In these few cases, reasonable Z_i values were filled in just to make the tables complete.

The Z_i can be calculated by using the defining equations given previously. Values are obtained at each mesh point in the grid of temperature-density arguments except those for which saturation occurs. FLUID does not attempt to calculate the Z_i for saturation conditions, but the interpolation algorithm requires that values be there. For this purpose, the Z_i tables under the saturation line can be filled in for each density argument by extrapolating downward in temperature from the unsaturated region.

Viscosity and conductivity. - In addition to the model-correction factors, the t-d tables can also be used for transport properties. In the oxygen tables, viscosity and conductivity values have been included. For these, no model is assumed. Instead, the two-dimensional interpolation in t-d gives directly the desired property. Values of viscosity and conductivity obtained from GASP were used for FLUID's tables. The units of viscosity are g/cm-sec and the units of conductivity are W/cm-K.

PROGRAM PERFORMANCE AND SIZE

Central Processing Unit (CPU) times required to complete one calculation with FLUID depend on the entry type and on the number of additional properties called for. These times have been measured on the Lewis Research Center's IBM 360 computer system as (typically) 4 milliseconds for T,ρ entries, 16 milliseconds for P,ρ entries, 12 milliseconds for T,P entries, and 50 milliseconds for P,S and P,H entries. Additional properties (NP > 1) require 1 millisecond each. These times are nearly independent of data-table size. The CPU time required by an initialization call to FLUID is insignificant.

One of the goals of the implementation of FLUID was to minimize program size. The Lewis Research Center's IBM 360 computer is a virtual-memory, time-share system. In this kind of system, small program size makes more CPU time available in real (connect) time. The compiled code for the FLUID subroutine and the two subroutines it references, NTRP and CUBIC, require 4141 (4-byte) words of storage.

The compiled code for the oxygen tables, developed in the manner described here and implemented as a subroutine, requires a total of 2960 words of storage. These tables provide the real-gas equation of state: the thermodynamic properties S, H, C_v, C_p; and sonic velocity, viscosity, and thermal conductivity. The tables are sufficiently fine to provide an agreement with the data base, which is, for the most part, better than

16

1 part per thousand. Tables for other gases may be expected to fall within a factor of 2 of this number of storage words for a comparable accuracy.

Appendix B lists a sample program that demonstrates the use of FLUID with the oxygen tables. It can be used to obtain thermodynamic properties in units other than those for which the data tables for FLUID were originally prepared. For example, by using the constants built into the program, values will be obtained that may be compared directly with the oxygen values in reference 5.

Lewis Research Center,
    National Aeronautics and Space Administration,
        Cleveland, Ohio, May 5, 1977,
            505-01.

APPENDIX A

ABSTRACT OF ENTRY POINTS IN FLUID

```
      SUBROUTINE FLUID0 (G,NG,GG,TT,DD,N1,N2,N3,SAT,NS,C,L)
C.....INTITIALIZING ENTRY FOR FLUID-PROPERTIES PACKAGE.  SETS UP TABLE
C     POINTERS FOR A PARTICULAR FLUID.  SUBSEQUENT CALLS (TO FLUID) USE
C     THESE POINTERS.

C.....CALLING ARGUMENTS:
C        G = (NON-DIMENSIONAL) TABLES IN TEMPERATURE ONLY.
C          G(1,1) = ARGUMENT VALUES (TEMPERATURE).
C          G(1,2) = ENTROPY, S0.
C          G(1,3) = ENERGY, U0.
C          G(1,4) = SPECIFIC HEAT AT CONSTANT VOLUME, CV0.
C        NG = LENGTH OF EACH G TABLE.
C        GG = TABLES IN TEMPERATURE AND DENSITY.
C          GG(1,1,1) = Z1, THE MODEL-DEPARTURE FACTOR.
C          GG(1,1,2) = Z2 FACTOR FOR ENTROPY (S).
C          GG(1,1,3) = Z3 FACTOR FOR ENTHALPY (H).
C          GG(1,1,4) = Z4 FACTOR FOR SPECIFIC HEAT (CV).
C          GG(1,1,5) = Z5 FACTOR FOR SPECIFIC HEAT (CP).
C          GG(1,1,6) = Z6 FACTOR FOR VELOCITY OF SOUND.
C          GG(1,1,N) = OTHER FACTORS (EG. TRANSPORT PROPERTIES).
C        TT = TEMPERATURE ARGUMENTS (REDUCED) FOR GG TABLES.
C        DD = DENSITY ARGUMENTS (REDUCED) FOR GG TABLES.
C        N1 = LENGTH OF TT TABLE AND FIRST DIMENSION OF GG TABLE.
C        N2 = LENGTH OF DD TABLE AND SECOND DIMENSION OF GG TABLE.
C        N3 = THIRD DIMENSION (NUMBER OF) GG TABLES.
C        SAT = (NON-DIMENSIONAL) SATURATION TABLES.
C          SAT(1,1) = SATURATION TEMPERATURE.
C          SAT(1,2) = SATURATION PRESSURE.
C          SAT(1,3) = SATURATED LIQUID DENSITY.
C          SAT(1,4) = SATURATED GAS DENSITY.
C          SAT(1,5) = SATURATED LIQUID ENTROPY.
C          SAT(1,6) = SATURATED GAS ENTROPY.
C          SAT(1,7) = SATURATED LIQUID ENTHALPY.
C          SAT(1,8) = SATURATED GAS ENTHALPY.
C        NS = LENGTH OF EACH SATURATION TABLE.
C        C = MISCELLANEOUS CONSTANTS:
C          C(1) = SPECIFIC GAS CONSTANT, R.
C          C(2) = CRITICAL TEMPERATURE, TC.
C          C(3) = CRITICAL PRESSURE, PC.
C          C(4) = CRITICAL DENSITY, DC (OR PSEUDO VALUE FOR BEST RANGE).
C          C(5) = CONVERGENCE REQUIREMENT FOR TEMPERATURE.
C          C(6) = CONVERGENCE REQUIREMENT FOR DENSITY.
C          C(7) = CONVERGENCE REQUIREMENT FOR ENTROPY.
C          C(8) = CONVERGENCE REQUIREMENT FOR ENTHALPY.
C        L = MEMORY (4 WORDS) FOR TABLE INDICES.

C.....VARIABLES IN COMMON BLOCK /FLUIDC/:
C        GAMMA = RATIO OF SPECIFIC HEATS, CP/CV.
C        WL = MASS-FRACTION IN THE LIQUID PHASE.
C        WG = MASS-FRACTION IN THE GAS PHASE.
C        DENSL = DENSITY OF SATURATED LIQUID,   SAME UNITS AS DC.
C        DENSG = DENSITY OF SATURATED GAS,      SAME UNITS AS DC.
C        ENTL = ENTROPY OF SATURATED LIQUID,    SAME UNITS AS R.
```

```
C          ENTG = ENTROPY OF SATURATED GAS,       SAME UNITS AS R.
C          ENTHL = ENTHALPY OF SATURATED LIQUID, SAME UNITS AS R*TC.
C          ENTHG = ENTHALPY OF SATURATED GAS,     SAME UNITS AS R*TC.


          COMMON /FLUIDC/ GAMMA,WL,WG,DENSL,DENSG,ENTL,ENTG,ENTHL,ENTHG



          ENTRY FLUID (TEMP,PRES,DENS,PROPS,NP,ENTRY,VAPOR,ERROR)
C.....MAIN ENTRY FOR PROPERTY CALCULATIONS.

C.....CALLING ARGUMENTS:
C          TEMP = FLUID TEMPERATURE,  SAME UNITS AS TC.
C          PRES = FLUID PRESSURE,  SAME UNITS AS PC.
C          DENS = FLUID DENSITY,  SAME UNITS AS DC.
C          PROPS = OTHER FLUID PROPERTIES:
C            PROPS(1) = COMPRESSIBILITY, PRES/(R*DENS*TEMP).
C            PROPS(2) = ENTROPY,  SAME UNITS AS R.
C            PROPS(3) = ENTHALPY,  SAME UNITS AS R*TC.
C            PROPS(4) = SPECIFIC HEAT, CV,  SAME UNITS AS R.
C            PROPS(5) = SPECIFIC HEAT, CP,  SAME UNITS AS R.
C            PROPS(6) = SONIC VELOCITY,  SAME UNITS AS SQRT(R*TC).
C          NP = NUMBER OF PROPERTIES TO BE CALCULATED.
C          ENTRY = INTEGER THAT SPECIFIES WHICH VARIABLES ARE INPUT:
C                = 1 IF TEMPERATURE AND DENSITY ARE GIVEN.
C                = 2 IF PRESSURE AND DENSITY ARE GIVEN.
C                = 3 IF TEMPERATURE AND PRESSURE ARE GIVEN.
C                = 4 IF PRESSURE AND ENTROPY ARE GIVEN.
C                = 5 IF PRESSURE AND ENTHALPY ARE GIVEN.
C          VAPOR = .TRUE. IF THE FLUID IS SATURATED.  IN THAT CASE,
C                  VALUES OF LIQUID AND GAS PHASES ARE GIVEN IN THE
C                  COMMON BLOCK /FLUIDC/.
C          ERROR = ERROR FLAGS (BITS -- LEAST SIGNIFICANT = 1):
C                  ** IF ERROR = 0,  ALL IS WELL **
C            BIT 1 = OUT OF RANGE IN SAT TABLE.
C            BIT 2 = OUT OF RANGE IN G TABLE.
C            BIT 3 = OUT OF RANGE IN TT TABLE.
C            BIT 4 = OUT OF RANGE IN DD TABLE.
C            BIT 5 = CONVERGENCE NOT ACHIEVED IN SOLVING INVERSE FUNCTION.



          ENTRY FLUIDL (PROPS,NP,ERROR)
C.....ENTRY FOR GETTING LIQUID-PHASE PROPERTIES NOT IN COMMON.



          ENTRY FLUIDG (PROPS,NP,ERROR)
C.....ENTRY FOR GETTING GAS-PHASE PROPERTIES NOT IN COMMON.
```

APPENDIX B

SOURCE-CODE LISTING (IN SFTRAN) OF A TEST PROGRAM FOR

OBTAINING OXYGEN VALUES FROM FLUID PACKAGE

```
C.....TESTS FLUID SUBPROGRAM AND OXYGEN TABLES.

      INCLUDE (TYPE STATEMENTS, ETC.)
      INCLUDE (CONSTANTS FOR COMPARING OXYGEN RESULTS TO NBS 564)

      NAMELIST /INPUT/ TEMP,PRES,DENS,ENT,ENTH,ENTRY,REF,NP,T0,P0,
     *                 TCONV,PCONV,DCONV,SCONV,HCONV,CCONV,ACONV,
     *                 SREF,HREF

C.....MAIN PROGRAM FLOW.
      CALL O2
      DO WITH
         READ (5,INPUT,DONE=END)
      UNTIL (DONE)
         IF (ENTRY.GE.1 .AND. ENTRY.LE.5) THEN
            DO CASE (ENTRY,5)
            CASE 1
               DO (TEMP AND DENS INPUT)
            CASE 2
               DO (PRES AND DENS INPUT)
            CASE 3
               DO (TEMP AND PRES INPUT)
            CASE 4
               DO (PRES AND ENT INPUT)
            CASE 5
               DO (PRES AND ENTH INPUT)
            END
            IF (VAPOR) THEN
               WRITE (6,1) TEMP,PRES,DENS,ENT,ENTH
            ELSE
               WRITE (6,2) TEMP,PRES,DENS,ENT,ENTH,CV,CP,A
               IF (NP.GT.6)  WRITE (6,3) (PROPS(I),I=7,NP)
            END
         ELSE
            IF (VAPOR) THEN
               IF (ENTRY.EQ.6) THEN
                  DO (LIQUID-PHASE VALUES)
               ELSE
                  IF (ENTRY.EQ.7) THEN
                     DO (GAS-PHASE VALUES)
                  ELSE
                     WRITE (6,4)
                  END
               END
            ELSE
               WRITE (6,4)
            END
         END
      END
      STOP
```

```
                 DEFINITION (CONSTANTS FOR COMPARING OXYGEN RESULTS TO NBS 564)

C        FOR TEMPERATURE IN K, PRESSURE IN ATMOSPHERES, REDUCED DENSITY,
C        ENTROPY AS S/R, ENTHALPY AS H/(R*T0), SPECIFIC HEATS AS CV/R AND
C        CP/R, AND REDUCED SONIC VELOCITY.   (THE VALUES OF SREF AND HREF
C        HERE GIVE AGREEMENT BETWEEN FLUID AND NBS 564 AT 273 K, 1 ATM.)

                 DATA T0,P0,TCONV,PCONV / 0.0, 0.0, 1.0, 9.8692 /,
       *              DCONV,SCONV,HCONV,CCONV / 699.8, 3.8487, 0.01409, 3.8487 /,
       *              ACONV,SREF,HREF,NP / 0.10045, 1.180, 408.14, 6 /
             END


C.....PROCEDURES.

             PROCEDURE (TEMP AND DENS INPUT)
                 TSAVE=TEMP
                 DSAVE=DENS
                 DO (GET ALL PROPERTIES VIA FLUID)
                 TEMP=TSAVE
                 DENS=DSAVE
             END

             PROCEDURE (PRES AND DENS INPUT)
                 PSAVE=PRES
                 DSAVE=DENS
                 DO (GET ALL PROPERTIES VIA FLUID)
                 PRES=PSAVE
                 DENS=DSAVE
             END

             PROCEDURE (TEMP AND PRES INPUT)
                 TSAVE=TEMP
                 PSAVE=PRES
                 DO (GET ALL PROPERTIES VIA FLUID)
                 TEMP=TSAVE
                 PRES=PSAVE
             END

             PROCEDURE (PRES AND ENT INPUT)
                 PSAVE=PRES
                 SSAVE=ENT
                 DO (GET ALL PROPERTIES VIA FLUID)
                 PRES=PSAVE
                 ENT=SSAVE
             END

             PROCEDURE (PRES AND ENTH INPUT)
                 PSAVE=PRES
                 HSAVE=ENTH
                 DO (GET ALL PROPERTIES VIA FLUID)
                 PRES=PSAVE
                 ENTH=HSAVE
             END

             PROCEDURE (GET ALL PROPERTIES VIA FLUID)
                 T=(TEMP+T0)/TCONV
                 P=(PRES+P0)/PCONV
                 D=DENS/DCONV
```

```
                 S=ENT/SCONV + SREF
                 H=ENTH/HCONV + HREF
                 CALL FLUID (T,P,D,PROPS,NP,ENTRY,VAPOR,ERROR)
                 IF (ERROR.NE.0) WRITE (6,5) ERROR
                 IF (REF) THEN
                     SREF=S
                     HREF=H
                     REF=.FALSE.
                 END
                 TEMP=T*TCONV - T0
                 PRES=P*PCONV - P0
                 DENS=D*DCONV
                 ENT=(S-SREF)*SCONV
                 ENTH=(H-HREF)*HCONV
                 IF (VAPOR) THEN
                     DENSL=DL*DCONV
                     DENSG=DG*DCONV
                     ENTL=(SL-SREF)*SCONV
                     ENTG=(SG-SREF)*SCONV
                     ENTHL=(HL-HREF)*HCONV
                     ENTHG=(HG-HREF)*HCONV
                 ELSE
                     CV=CV*CCONV
                     CP=CP*CCONV
                     A=A*ACONV
                 END
             END
         END

         PROCEDURE (LIQUID-PHASE VALUES)
             CALL FLUIDL (PROPS,NP,ERROR)
             DO (REPORT PHASE VALUES)
         END

         PROCEDURE (GAS-PHASE VALUES)
             CALL FLUIDG (PROPS,NP,ERROR)
             DO (REPORT PHASE VALUES)
         END

         PROCEDURE (REPORT PHASE VALUES)
             IF (ERROR.NE.0) WRITE (6,5) ERROR
             CV=CV*CCONV
             CP=CP*CCONV
             A=A*ACONV
             IF (ENTRY.EQ.6) THEN
                 WRITE (6,6) DENSL,ENTL,ENTHL,CV,CP,A
             ELSE
                 WRITE (6,7) DENSG,ENTG,ENTHG,CV,CP,A
             END
             IF (NP.GT.6) WRITE (6,3) (PROPS(I),I=7,NP)
         END

  1      FORMAT (' T,P,D,S,H:',5(1PE13.4))
  2      FORMAT (' T,P,D,S,H,CV,CP,A:',8(1PE13.4))
  3      FORMAT ('   ADDITIONAL PROPS:',8(1PE13.4))
  4      FORMAT (' *ERROR* BAD VALUE FOR ENTRY')
  5      FORMAT (' ERROR =',I3)
  6      FORMAT (' DL,SL,HL,CV,CP,A:  ',6(1PE13.4))
  7      FORMAT (' DG,SG,HG,CV,CP,A:  ',6(1PE13.4))
```

```
      DEFINITION (TYPE STATEMENTS, ETC.)
          INTEGER ENTRY,ERROR
          LOGICAL VAPOR,REF,DONE
          DIMENSION PROPS(6)
          COMMON /FLUIDC/ GAMMA,WL,WG,DL,DG,SL,SG,HL,HG
          EQUIVALENCE (Z, PROPS(1)), (S, PROPS(2)), (H, PROPS(3)),
      *             (CV,PROPS(4)), (CP,PROPS(5)), (A, PROPS(6))
      END

      END
```

REFERENCES

1. Hendricks, Robert C.; Baron, Anne K.; and Peller, Ildiko, C.: GASP - A Computer Code for Calculating the Thermodynamic and Transport Properties for Ten Fluids: Parahydrogen, Helium, Neon, Methane, Nitrogen, Carbon Monoxide, Oxygen, Fluorine, Argon, and Carbon Dioxide. NASA TN D-7808, 1975.

2. Hendricks, Robert C.; Peller, Ildiko C.; and Baron, Anne K.: WASP - A Flexible FORTRAN IV Computer Code for Calculating Water and Steam Properties. NASA TN D-7391, 1973.

3. Rowlinson, J. S.: The Properties of Real Gases. Handbuch der Physik. S. Flugge, ed., Vol. 12, Springer (Berlin), 1958, pp. 1-72.

4. Fessler, Theodore E.; and Ford, William F.: User's Guide for SFTRAN/360. NASA TP-1006, 1977.

5. Hilsenrath, Joseph; et al.: Tables of Thermal Properties of Gases Comprising Tables of Thermodynamic and Transport Properties of Air, Argon, Carbon Dioxide, Carbon Monoxide, Hydrogen, Nitrogen, Oxygen, and Steam. Circular 564, Nat. Bureau Std., 1955.